UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/054,544	01/18/2002	Joerg Jahnke	P-5513	7147

24209 7590 11/01/2007

GUNNISON MCKAY & HODGSON, LLP
1900 GARDEN ROAD
SUITE 220
MONTEREY, CA 93940

EXAMINER	
PANNALA, SATHYANARAYAN R	
ART UNIT	PAPER NUMBER
2164	

MAIL DATE	DELIVERY MODE
11/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/054,544
Filing Date: January 18, 2002
Appellant(s): JAHNKE ET AL.

Forrest Gunnison
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed on 7/23/2007 appealing from the Office action mailed on 7/20/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal. Examiner is relied upon the appellant's statement contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

In response to the Final Office Action mailed on 7/20/2006 Appellant's Amendment filed on 9/6/2006. This amendment was not entered.

(5) Summary of Claimed Subject Matter

The summary of invention contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Chau et al. (US Patent 6,721,727) hereinafter Chau

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section

351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-33 are rejected under 35 U.S.C. 102(e) as being anticipated by Chau et al. (US Patent 6,721,727) hereinafter Chau.

3. As per independent claim 1, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID

and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (Fig. 3, col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

4. As per dependent claim 2, Chau teaches the claimed step of "element data set includes character data" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

5. As per dependent claim 3, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

6. As per dependent claim 4, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

7. As per dependent claims 5, Chau teaches the claimed step of "element data set includes an element name" as the user decides how XML document data is to be accessed in a database. The user defines a document access definition (DAD) as an element (col. 12, lines 61-663).

8. As per dependent claim 6, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as the column of the side table contains the value of a location path of the specified type. Name of the column is the alias name of the location path which identifies an element (Fig. 3, col. 13, lines 64-67).

9. As per dependent claim 7, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an

attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the column of the side table contains the value of a location path of the specified type. Name of the column is the alias name of the location path which identifies an element or attribute (Fig. 3, col. 13, lines 64-67).

10. As per dependent claim 8, Chau teaches the claimed step of "attribute data set includes an attribute name" as the attribute is the name of an XML element and it is the tag name. This has the unique and it is adopted form XPTH (col. 15, lines 35-38).

11. As per dependent claim 9, Chau teaches the claimed step of "attribute data set includes an attribute value" as multiple-occurring element text or attribute value when generating XML documents (col. 14, lines 65-67).

12. As per dependent claim 10, Chau teaches the claimed step of "attribute data set includes an attribute value" as multiple-occurring element text or attribute value when generating XML documents (col. 14, lines 65-67).

13. As per dependent claim 11, Chau teaches the claimed step of "the markup document is an XML document" as extensible markup language is for creating XML documents (col. 2, lines 20-22).

14. As per independent claim 12, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3; col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an

element to which the one attribute is assigned" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

Further, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as by storage, the XML system provides mechanisms for storing and retrieving XML documents in relational database. The DB2 XML extender 200 takes an XML document 206 as the input, stores XML document 206 in DB2 210 either internally inside DB2 210 or externally on the files system as one or more XML files 208. Chau teaches two storage techniques and they are Xcolumn defines how to store and retrieve entire XML documents as column data of the XML user defined type and this method allows storing of elements and attribute values (Fig. 2, col. 5, lines 40-42; col. 6, lines 6-12; col. 7, lines 52-58; col. 8, lines 2-25 and col. 19, lines 28-36).

Finally, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data

set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the relational side tables are created for indexing elements or attributes of documents stored in an XML column. Creating number of side tables is based on the understating of the DTD and XML documents, the application table 300 has a root_id in common with each side table302, 304, 306 and 308. Every side table will have a unique attribute for an order_tab side table has order_key as the primary key (Fig. 3, col. 13, lines 57-59; col. 18, line 67 to col. 19, line 3; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

15. As per dependent claim 13, Chau teaches the claimed step of "element data set includes character data" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

16. As per dependent claim 14, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

17. As per dependent claim 15, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

18. As per independent claim 16, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "an element table wherein said element table is configured to store a plurality of element records corresponding to a plurality of elements of a markup document so that said relational database includes a plurality of element records, and further wherein each element record includes an assigned element ID field and an element data set field" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed "an attribute table wherein said attribute table is configured to store a plurality of attribute records corresponding to a plurality of

attributes of said markup document so that said relational database includes a plurality of attribute records, and further wherein each attribute data record includes an element ID field and an attribute data set wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

19. As per dependent claim 17, Chau teaches the claimed "the element data set includes a character data field" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

20. As per dependent claim 18, Chau teaches the claimed "the element data set includes a parent element ID field" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

21. As per dependent claims 19, Chau teaches the claimed "the element data set includes an element number field" as the invoice_number is a primary key and examiner interpreted as an element number (Fig. 3, col. 19, lines 31-35).

22. As per dependent claim 20, Chau teaches the claimed "element data set includes an element name field" as the sales_person name is interpreted as element name (Fig. 3, col. 19, lines 30-31).

23. As per dependent claims 21, Chau teaches the claimed "the element data set comprises an element name ID field" as the invoice_number is a primary key and examiner interpreted as an element name ID or the sales_person ID column can created in the sales_tab table (Fig. 3, col. 19, lines 31-35).

24. As per dependent claim 22, Chau teaches the claimed "an element name table wherein said element name table is configured to store a plurality of element name records, and further wherein each element name record includes an element name ID field and a corresponding element name field" as the application table is the same as the element table (col. 8, lines 53-56).

25. As per dependent claim 23, Chau teaches the claimed "attribute data set includes an attribute name and an attribute value" as the term beginning with ATTLIST refer to attributes of an XML document as listed in the LineItem.dtd and relational tables

created for indexing elements or attributes of documents stored in an XML column and the table is specified by name, type, path and etc. (col. 11, lines 61-62 and col. 13, line 57 to col. 14, line 6).

26. As per dependent claim 24, Chau teaches the claimed "attribute data set contains an attribute name ID" as side tables will have attribute ID, for example see the side_table order_tab has order_key, which is interpreted as an attribute name ID (Fig. 3, col. 21, lines 26-30).

27. As per dependent claim 25, Chau teaches the claimed "an attribute name table wherein said attribute name table is configured to store a plurality of attribute name records wherein each attribute name record includes an attribute name ID field and a corresponding attribute name field" as the side tables will be the same as attributes table and the could have an attribute name, attribute name ID and etc., for example see the side_table order_tab has order_key (Fig. 3, col. 21, lines 26-30).

28. As per independent claim 26, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space

and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "storing an element record for every element of a plurality of elements of said markup document in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed "storing an attribute record for every attribute of a plurality of attributes of said markup document in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute

table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (Fig. 3, col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

29. The computer program product of claim 27, Chau teaches the claimed "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

30. The computer program product of claim 28, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element

table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

31. The computer program product of claim 29, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

32. As per independent claim 30, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "a memory having stored therein a module for transferring data from a markup document into a relational database" as the stored procedures and modules are code organization to compose XML documents (Fig. 7, col. 42, lines 42-53).

Further, Chau teaches the claimed "a processor coupled to said memory wherein execution of said module" as a server computer 104 executing software and other computer programs and to connect the server system 104 to data sources 106 (Fig. 1, col. 4, lines 3-5).

Further, Chau teaches the claimed "storing an element record for every element of a plurality of elements of said markup document in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Finally, Chau teaches the claimed "storing an attribute record for every attribute of a plurality of attributes of said markup document in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element

table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

33. As per dependent claim 31, Chau teaches the claimed "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

34. As per dependent claim 32, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

35. As per dependent claim 33, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

(10) Response to Argument

A. Whether Claims 1, 2, 5, 8 to 11, 16, 17, 19 to 21, 23, 24, 26, and 30 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,721,727 of Chau, hereinafter referred to as Chau?

Appellant argument regarding claim 1, first limitation, page 15, lines 24-27, stated as "storing an entire XML document with a primary key in the Application Table teaches away from storing a plurality of elements records for a single markup document in an element table."

In response to the Appellant's argument, examiner respectfully, disagrees and details as follows:

a) Regarding Claim 1, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. **A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store on or more elements or attributes. The side tables are used to locate the data in the main table.** A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62- 67, lines 3-5 and 11-12).

The first limitation:

b) Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said

relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further response to Appellant's argument, examiner would like to explain gain that Chau teaches as XML enables storing entire XML documents into a database. Chau explained with an example in the Fig. 3. The root_id in the application table (see Fig. 3, Sales_Tab, 300) is unique element ID and the user creates root_id as a primary key of the application table.

In response, a person of ordinary skill in the art will agree that a primary key in a table is a **unique key** and it is formed by one or more elements in the table. Chou has clearly teaches that **a main table** is created having a column for storing **a document** and it **has one or more elements or attributes**. One or more **side tables** are created to store on or more elements or attributes. The side tables are used **to locate the data** in the main table. This means even though document is in one column, the side tables facilitate required elements from the document. Chau also further teaches that when there is no primary key in the table, then **XML system create**s a primary key as DXXROOT_ID and **all side tables will have this key**.

Further, **Appellant argued, page 16, lines 9-10, as** "This general statement teaches nothing concerning how the XML documents and the root ids are stored in the application table."

In response to Appellant's argument, Examiner states that the **procedure** of **storing XML and primary key** is **not in the limitation** and moot to respond.

Further, **Appellant argued, page 17, last paragraph, as** "The element records in Claim 1 are rows populated with unique element IDs and element data sets from a single markup document. There is a different element record for each element in the plurality of elements from the single markup document."

In response, a person of ordinary skill in the art will agree that every database will have one or more tables and each table with rows and columns. As per Chau, every row will have a primary key column with a unique element, either created by user or created by XML system. MPEP 2111 states that claims must be given their broadest reasonable interpretation.

Further, **Appellant argued, page 18, lines 25-27, as** "A single record for an XML document as in the Application Table fails to teach a plurality of records of any type for that XML document."

In response, a person of ordinary skill in the art will agree that every database will have one or more tables and each table contains rows and columns. Applicant is interpreting each table of Chau teaching contains only one, which is not true. Each row

of an Application (as example in Fig. 3, Sales Table). It has multiple columns and multiple rows and each row will have a XML document. Chau never stated that each table contains only one document.

Further, **Appellant argued, page 19, last paragraph, as** "Claim 1 recites precisely how many element records are stored in the element table, for example, "an element record for every element of said plurality of elements."

In response, claim 1 recites as plurality of element records are stored in the element table and Chau also state the same as plurality of element records are stored in the application table (300). A table is used to store several records and each record having a unique key as primary key.

Further, **Appellant argued, page 22, last paragraph, as** "the use of the same root id in a side table for attributes from a XML document not only fails to teach exactly, the record dependent element IDs recited in Claim 1, but also teaches away from such IDs. This is additional evidence that the anticipation rejection is not well founded."

In response, Chau teaches as "FIG. 3 illustrates an application or main table and its four side tables. The Application table 300 has a root_id in common with each side table 302, 304, 306, and 308 (Fig. 3, col. 18, line 66 to col. 19, line 3). Further Chau teaches as "when side tables are created, they are tied together with the main (or application) table through the notion of root_id. A user can decide whether the primary key of the application table is to be the "root_id". If the primary key does not exist in the

application table, or for some reason a user doesn't want to use the primary key, then XML System will alter application table to add a column DXXROOT_ID for storing a unique identifier created at insertion time."

The second limitation:

c) Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (Fig. 3, col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).
Claims 2, 5, 8 to 11 are dependent on claim 1. Claim 16 is parallel to claim 1 and claims 17, 19 to 21, 23, and 24 are dependent. Claim 26 and are parallel to claim 1.

Further, **in response to Appellant's argument**, a recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim. Therefore, the prior art of Chau teaches the claims 1, 2, 5, 8 to 11, 16, 17, 19 to 21, 23, 24, 26, and 30 and the rejection should be sustained.

B. Whether Claims 3, 4, and 18 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,721,727 of Chau?

Appellant argument regarding claim 3, Chau fails to teach the invention to the same level of detail as recited in the claim, and so the anticipation rejection is not well founded

In response to the Appellant's argument, examiner respectfully, disagrees and details as follows:

a) As per **dependent claim 3**, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-56).

b) Chau teaches in similar to the current invention with Fig. 1A, as when side tables are created, they are tied together with the main (or application) table through the notion of root_id. A user can decide whether the primary key of the application table is to be the "root_id". Element table and attribute table contains same element ID which is similar to application table and side tables having root_id.

c) Claim 3 is dependent on claim 1. As Appellant stated that claims 4 and 18 recite a limitation equivalent to that in claim 3. Claim 4 is dependent on claim 1 and claim 18 is dependent on claim 16.

C. Whether Claims 6, 22, 27, 31 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,721,727 of Chau?

Appellant argument regarding claim 6 as "the cited portion of the DAD of Chau fails to mention storing for 'every unique element name of the plurality of elements.'"

In response to the Appellant's argument, examiner respectfully, disagrees and details as follows:

a) As per **dependent claim 6**, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as the column of the side table contains the value of a

location path of the specified type. Name of the column is the alias name of the location path which identifies an element (Fig. 3, col. 13, lines 64-67).

b) Chau teaches similar to current invention as shown in the Fig. 1A, as the column of the side table, which contains the value of a location path of the specified type. name: name of the column. It is the alias name of the location path, which identifies an element or attribute, type: the data type of the column. It can be any SQL data type. path: the location path of an XML element or attribute.

c) Claim 6 is dependent on claim 1. Clams 22, 27 and 31 recite equivalent to that in claim 6. Claim 22 depends on claim 16, claim 27 depends on claim 26 and claim 31 depends on claim 30.

D. Whether Claims 7, 25, 28, 32 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,721,727 of Chau?

Appellant argument page 29, 2nd paragraph from the bottom of the page, regarding claim 7 as "Further, the cited portion of the DAD of Chau fails to mention storing for 'every unique attribute name of the plurality of attributes.' It also does not teach the attribute name record or a unique attribute name ID."

In response to the Appellant's argument, examiner respectfully, disagrees and details as follows:

a) As per dependent claim 7, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the column of the side table contains the value of a location path of the specified type. Name of the column is the alias name of the location path which identifies an element or attribute (Fig. 3, col. 13, lines 64-67).

b) Chau teaches similar to current invention as the column of the side table, which contains the value of a location path of the specified type. name: name of the column. It is the alias name of the location path, which identifies an element or attribute, type: the data type of the column. It can be any SQL data type. path: the location path of an XML element or attribute.

c) Claim 7 is dependent on claim 1. Clams 25, 28 and 32 recite equivalent to that in claim 7. Claim 25 depends on claim 16, claim 28 depends on claim 26 and claim 32 depends on claim 30.

E. Whether Claims 12 to 15, 29 and 33 are unpatentable under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 6,721,727 of Chau

Appellant argument page 31, 2nd paragraph from the top of the page as Appellant notes that Claim 12 includes the combination of Claims 1, 6 and 7. The

rejection of Claim 12 includes the same errors as noted above with respect to Claim 1 and so the remarks for Claim 1 are incorporated herein by reference instead of repeating them in the context of Claim 12.

In response to the Appellant's argument, examiner respectfully, disagrees and details as follows:

a) As per **independent claim 12**, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is

no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

Further, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as by storage, the XML system provides mechanisms for storing and retrieving XML documents in relational database. The DB2 XML extender 200 takes an XML document 206 as the input, stores XML document 206 in DB2 210 either internally inside DB2 210 or externally on the files system as one or more XML files 208. Chau teaches two storage techniques and they are Xcolumn defines how to store and retrieve entire XML documents as column data of the XML user defined type and this

method allows storing of elements and attribute values (Fig. 2, col. 5, lines 40-42; col. 6, lines 6-12; col. 7, lines 52-58; col. 8, lines 2-25 and col. 19, lines 28-36).

Finally, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the relational side tables are created for indexing elements or attributes of documents stored in an XML column. Creating number of side tables is based on the understating of the DTD and XML documents, the application table 300 has a root_id in common with each side table302, 304, 306 and 308. Every side table will have a unique attribute for an order_tab side table has order_key as the primary key (Fig. 3, col. 13, lines 57-59; col. 18, line 67 to col. 19, line 3; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

b) Claim 12 is an independent claim. Claims 13-15 depends on claim 12, claim 29 depends on claim 26 and claim 33 depends on claim 30.

Conclusion

The reference discloses the claimed invention of storing data of an XML-document in a relational database. Chau teaches XML document stored as column data in very details and most of the claims and their limitations. Examiner strongly feels that the reference used meets all claims and there is no justification of Appellant arguing that the storing data of an XML-document is different from XML document stored as column data.

For the above reasons, it is believed that the rejections should be sustained.

(11) Related Proceedings Appendix

No decision rendered by a court or the Board of Appeals and Interferences is identified by the examiner in the "Related Appeals and Interferences" section of this examiner's answer.

Respectfully Submitted,

Sathyanarayan Pannala
Primary Examiner

srp
October 24, 2007

Conferees:

1. Mohammad Ali, Supervisory Patent Examiner, Art Unit 2169

2. Charles Rones, Supervisory Patent Examiner, Art Unit 2164

CHARLES RONES
SUPERVISORY PATENT EXAMINER